VZ-LINK NEWLETER

<< AUCKLAND VZ 300/200 USERS CLUB >>

ISSUE # 21 JUNE 1988

Well here we are again,another month and information packed issue. Thank you to those people who attended the workshop/meeting on 28Th May.
To those that couldn't make it,The next one will be held on:- Saturday 27Th August. Write it down so you don't forget.
You may not learn anything new ,but you maybe able to teach other users something.

I would like to organise something for TELETHON.
If you have any ideas let me know.Perhaps a computer thon would be a super idea.

Dick Smiths are moving their commerce street store to 442 Karangahape Road opening on 30th May.They have just had a moving sale.
The VZ gear was really cheap,but there was not much of it.I managed to get a rtty kit and some software.

In the April issue of Australian Electronics Monthly there is a 7 page item on how to make and install a "Ultra-Graphics" adaptor for the VZ.If you can't get a copy,give me a call.
There Is also suppose to be a VZ Eprom programer project coming soon in ETI magazine.

There is still some membership renewals due.If you have forgotten please send them as soon as possible.

In this issue we have another fine item written by Bob Kitch.
Also a couple of Technical Bulletins which may help some of the newer users.
To top it all off we have another program from Danny Todd.

Check out page 14 before you clean your disk drive,you maybe Glad you did.

Don't forget your items for the newsletter.
Also when you send me programs,could you please include a letter giving me permission to use it in the newsletter.
I would also like permission to send your programs etc for the Australian newsletters to use.

Well,enough from me,so turn the page and read on.

24-BIT I/O PROGRAMMABLE PERIPHERAL INTERFACE BOARD FOR VZ200/300 COMPUTERS.

by R. B. Kitch.
7 Eurella St.,
KENMORE, QLD. 4069

A large number of Users require digital (switching) I/O from their VZ computer to allow interfacing to "real-world" devices. Robotic control and electric train set control are the most usually cited requirements, but house security, serial or parallel communications with other computers or printers, or synchronizing other peripherals are some possible uses of programmable ports.

The interface described herein provides, in its simplest configuration, twelve digital outputs and twelve digital inputs. The I/O lines are latched (thereby eliminating pull-up or pull-down resistors), buffered and seperately powered or isolated from the computer. The outputs may be interfaced to relays, DACs or LEDs. The inputs are usually switches or ADCs.

The interface uses a "smart" programmable port so that it can be readily reconfigured by suitable software. The Intel 82C55A PPI is used. It can also be programmed to look like three simple I/O ports (mode 0), two handshaking I/O ports (mode 1) or a bi-directional I/O port with five handshaking signals (mode 2). The modes can be mixed.

Address decoding of the four I/O ports required by the PPI on the Z80 bus structure is also provided. Decoding of the Z80's *IORQ, *WR and *RD control signals is also done on-board.

The advantage of using a "smart" interface is that the mode selection is readily selectable by one control word - no matter how complex the configuration may be. "Dumb" ports (for example 74LS244 and 74LS374 bus drivers) can however take a little more of a beating and have a higher current driving capacity. Their timing is often a little less critical than a smart port. The smart port which provides the greatest number of bits-per-dollar is the Intel 82C55A Programmable Peripheral Interface (PPI) which is the CMOS high speed version of the earlier 8255.

FUNCTIONAL DESCRIPTION.

The 82C55A is a general purpose I/O interfacing device providing 24 I/O lines. These are organised as three 8-bit I/O ports labelled A, B and C. Individual bits cannot be programmed as inputs or outputs. Instead, all of the bits in Port A or B are programmed as one byte. The four high- and four low-order bits of Port C can be programmed as two seperate nibbles however. This compares more than favourably with the Z80 PIO device. In addition, the 82C55A does not require to be connected to the system clock, M1 and interrupt lines as required by the Z80 PIO. The 82C55A matches the terminations available from the VZ I/O port located at the rear of machine.

The PPI may be used to interface peripheral equipment onto a microcomputer bus system. The functional configuration of the PPI is programmed by the system software via the Control or Status Word port, such that usually no external logic is necessary to interface peripheral devices or structures. All communication with the PPI occurs over the eight data lines. In fact, the PPI is actually seen as four I/O ports on the bi-directional data bus and can be mapped anywhere into address space by suitable decoding of the address lines A2-A7. The specific port address is controlled by the *CS input, which controls access to the PPI.

Address lines A0-A1 determine which of the four ports is addressed as follows-

Port A	addr. 0	A0=0	A1=0
Port B	addr. 1	A0=1	A1=0
Port C	addr. 2	A0=0	A1=1
Control	addr. 3	A0=1	A1=1

When *IORD is low, any of the three data ports can be read by applying the appropriate combination to A0 and A1. When both A0 and A1 are high, the Control Port is accessed. Note that this register can only be written to, not read from. When the PPI is not accessed (*CS high or, *WR and *RD high), the data bus connections are placed in a high impedance state.

Port A can be configured as-

i/. one 8-bit data output latch/buffer with 2.5mA DC drive capacity - enough to drive one standard TTL load or four LSTTL loads.

ii/. one 8-bit data input latch/buffer.

(Both pull-up and pull-down bus hold devices are present).

Port B can be configured as-

i/. one 8-bit input data output latch/buffer with 2.5mA DC drive capacity.

ii/. one 8-bit data buffer (no latching).

(Only pull-up bus hold devices are present).

Port C can be configured as-

i/. one 8-bit data output latch/buffer with 2.5mA DC drive capacity.

ii/. one 8-bit data input buffer (no latching).

iii/. two 4-bit latched ports under mode control.

Each port contains a 4-bit latch and can be used for control signal outputs and status signal inputs in conjunction with Ports A and B. Only pull-up bus hold devices are present.

The Control Port can be both written to and read from to achieve control. Note that the "asymmetry" of the port features can be used to advantage when designing I/O requirements with respect to latching/non-latching.

INTERFACING I/O INTO THE PPI.

The PPI can sink or source 2.5mA on output. This may be enough to drive low power devices, but generally to drive LEDs (20mA) or relays (20-100mA) higher current sourcing capacity is required. VLSI devices are also not terribly immune to noise or voltage spikes. It is generally safer therefore to interpose bus drivers/buffers. In the PPI configuration detailed herein, all latching of I/O data is taken care of in the PPI. Propagation delays introduced by the 74LS-series devices are not significant to a 4MHz Z80 system - particularly when the Z80 inserts an (extra) WAIT state into the timing of any read/write operation involving an I/O port. This feature of the Z80's timing allows "slow" peripheral devices time to latch data. This timing feature is the reason why it is better to map the PPI into I/O port address space rather than memory address space.

The data supplied indicates interface circuits for high and low voltage switching via a 74LS244 bus driver. Isolation is provided where necessary by opto-isolators or relays. A wide variety of circuit configurations are possible here depending upon what is being interfaced onto the bus.

SELECTING OPERATING MODE.

Once the PPI has been interfaced between the Z80 bus system and the "real-world" devices, the operating mode must be selected by configuring the Control Port. Only mode 0, for basic I/O is detailed herein. Basic I/O uses unconditional or non-handshaking features, and is best used when the peripherals can always be assumed to be ready.

A single control word is written to the Control Port.

This word is built up as follows-

Bit 7 =1 modes 0/1/2 to be selected (rather than bit set/reset mode).

Bit 6&5=00 mode 0 selected (01 for mode 1, 1X for mode 2).

Bit 4 =1 Port A - input (0 for output).
Bit 3 =0 Port C upper nibble - output (1 for input).
Bit 2 =0 mode 0 selected (1 for mode 1).
Bit 1 =0 Port B - output (1 for input).
Bit 0 =1 Port C lower nibble - input (0 for output).

The control word is 10010001B, 91H or 145D for the configuration shown here. It can be seen that the three ports are broken into two groups for mode selection. Port A and the high-order bits of Port C can be programmed for any of the modes 0 through 2. Port B and the low-order bits of Port C can be programmed in modes 0 and 1 only.

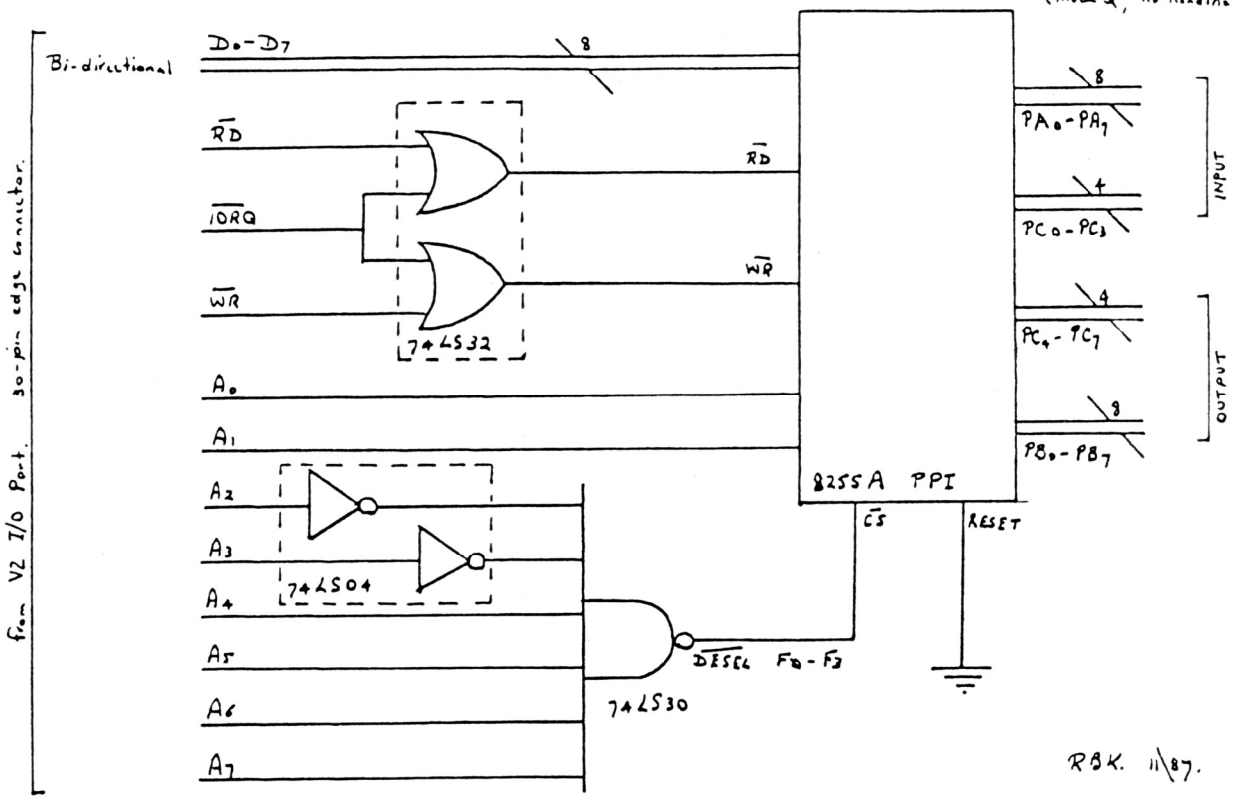
MULTIPLY
WRITTEN BY KEITH WRIGHT

```
1 REM MULTIPLYING
20 CLS
50 FORN=1TO128:PRINT"▚▚▚";:NEXT
60 PRINT@232,"[illegible]";
70 FORN=1TO3000:NEXT
71 CLS:PRINT:PRINT
75 INPUT"HELLO,WHAT'S YOUR NAME";N$
100 PRINT@200,"GLAD YOURE HERE,";N$
105 PRINT"LETS DO MULTIPLICATION"
107 PRINT"WHAT NUMBER TO START WITH?"
110 INPUT X:CLS
115 PRINT:PRINT:PRINT:PRINT"ALLRIGHT,WE'LL START WITH";X
117 FORW=0TO1500:NEXT
120 C=0:R=0:W=0
150 CLS:PRINT:PRINT:PRINT:Y=INT(RND(9)):PRINTX;"TIMES";Y;"=";
160 INPUTZ:IFX*Y=ZTHENPRINT"YES!!":SOUND31,1:SOUND29,1:GOTO200
165 IFRND(2)>1THENPRINT"NO!":ELSEPRINT"OOPS"
170 SOUND 4,2;2,4:W=W+1
180 GOTO160
200 R=R+1:C=C+1:IFC<10THEN150
220 PRINT:PRINT:PRINT;R;"CORRECT,"
230 PRINTW;"BAD GUESSES."
240 PRINT"  WHAT NUMBER WOULD YOU LIKE ":GOTO110
```

24 bit Programmable Peripheral Interface for V2 Computers

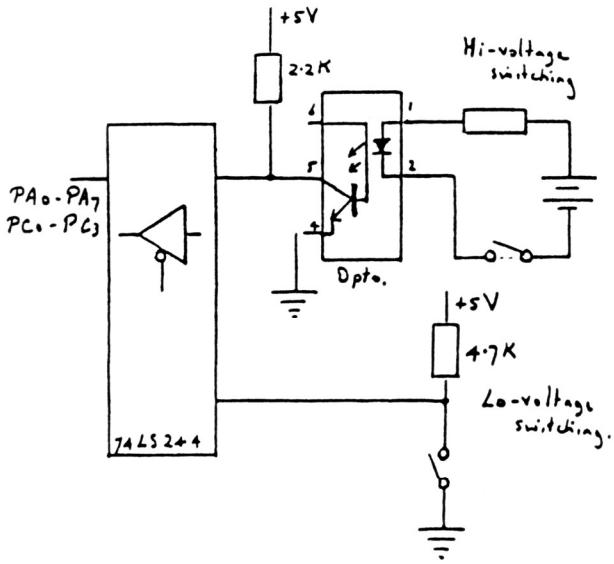
Configuration Details - as shown.

Port A	I/O addr.	F0 H	240D	input
Port B		F1 H	241D	output
Port C		F2 H	242D	- upper output - lower input.
Control.		F3 H	243D	- 91H, 145D, 10010001B (mode 0, no handshaking)



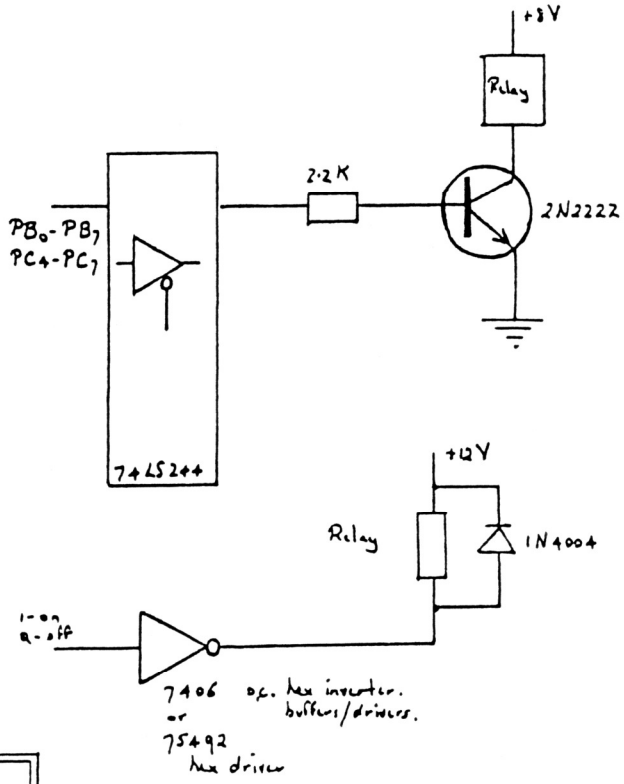
RBK. 11\87.

INPUT Interfaces.



74LS244 will handle directly 5 - 2.5 V switching.

OUTPUT Interfaces.



Interfacing Digital I/O to PPI

RBK 11\87.

VZ ASSEMBLY

Over the last few months,we've been looking at various ways of handling mathematical operations,using ASSEMBLY LANGUAGE.
After studying addition,subtraction, multiplication and division,it soon becomes clear that carrying out such processes can take some quite lengthy coding.It's quite obvious then that writing code for mathematical functions(square root,logs etc) and digital divides etc are well out of the scope for the hobbiest programmer.In such cases,it is necessary to make use of the code that already exists in the BASIC operating system.All that is then required is a knowledge of where the routines are in ROM,and how to use them.

Firstly,we need to understand how numbers are stored by BASIC. We will be looking at the three main types of' computer variable:- Integer,Single precision and Double precision.

If we think about what the BASIC system offers in terms of mathematical operations,it's clear that just about every type of operation is incorporated into the system.What's more,both integer and decimal numbers can be substituted.

Let's consider the integers first.Although it may not seem obvious,the BASIC system handles these separately to decimal numbers.The system can only operate in one mode at a time i.e integer,single or double precision.You can force any of the modes by using the DEFINT (define integer),DEFSNG (define single precision) or DEFDBL if you have extended BASIC.Otherwise,you can specify integer by using integers with % symbols..eg:A%.

So what's the difference between Integer,Single and Double precision?.
The difference is in how much memory the systems sets aside to store the number.When we're dealing with numbers,we must consider their magnitude(how large or small),their sign (+ or -) and the numbers of significant figures involved (i.e how 'precise' the number is).
Put simply,the larger the number,the more memory required to store it,and the more precise the number,the more memory required to store it.
This is basically the difference between Integer,Single and Double Precision.

Integer can only be used to store whole numbers.Two bytes are set aside,that is,16 bits.One bit (the most significant) is used to indicate the sign of the Integer (0=positive,1=negative).
The remaining 15 bits represent the actual magnitude of the number in binary.If you carry out a simple calculation (2^15-1),you'll see that the range of an Integer is -32767 to 32767
(sound familiar ?-this is the size of the arguement in a POKE or PEEK).
Clearly this is quite restrictive,but the computer can handle these numbers relatively quickly.

The next level of variable is Single Precision.In this and double precision,numbers are stored in 'scientific notation',i.e. in the form:

Ax10^b

Where A is called the mantissa, and B the exponent.24 bits are set aside for the mantissa (1 bit for sign,23 bits for value),and eight bits for the exponent.If you do the necessary calculations,you'll see that a single precision number can have about six significant figures eg: 327686 or 3.27686 (if you don't understand the term 'significant figures' consult a maths text).

CONTINUED ON NEXT PAGE

This is an improvement on Integer in that larger numbers and decimals are possible. However,we still don't have quite the size we would like.

Double Precision numbers require a total of 56 bits (8 bits exponent,48 bits signed mantissa),and have a precision of up to about 16 significant figures.

As mentioned earlier,the BASIC system can only work with numbers of the same type.Clearly,the size of single and double precision numbers means that they can not be handled as a whole by any of the Z80 registers.Thus,two areas of RAM are set aside in the Communications region,called respectively 'Working Register Area 1' (WRA 1) and 'Working Register Area 2'(WRA 2).
WRA 1 is located at 31005-31012 (791DH-7924H)
WRA 2 is located at 31015-31022 (7927H-792EH).Each area is set up as shown below:

ADDRESS:	FOR INTEGER:	FOR SINGLE PREC:	FOR DOUBLE PREC:
31005	-	-	LSB
31006	-	-	NMSB
31007	-	-	NMSB
31008	-	-	NMSB
31009	LSB	LSB	NMSB
31010	MSB	NMSB	NMSB
31011	-	MSB	MSB
31012	-	EXPONENT	EXPONENT

WRA 2 has the same format. (just change the addresses)
LSB = Least Significant Byte MSB = Most Significant Byte NMSB = Next Most Significant Byte

Since the system can only operate in one mode at any particular time, operations can only be carried out between the same type of variable. Since there are four mathematical operations (+,-,*,/) and three types of values, there must be 12 arithmetic routines:

TYPE:	OPERATION:	CALL ADDRESS:	SOURCE REG.:	DESTINATION:
INTEGER	Addition	0BD2H	HL + DE	HL
	Subtraction	0BC7H	HL - DE	HL
	Multiply	0BF2H	HL * DE	HL
	Divide	2490H	DE / HL	WRA1
SNGPREC	Addition	0716H	(BCDE)+WRA1	WRA1
	Subtraction	0713H	WRA1-(BCDE)	WRA1
	Multiply	0847H	WRA1*(BCDE)	WRA1
	Divide	08A2H	WRA1/(BCDE)	WRA1
DBLPREC	Addition	0C77H	WRA1+WRA2	WRA1
	Subtraction	0C70H	WRA1-WRA2	WRA1
	Multiply	0DA1H	WRA1*WRA2	WRA1
	Divide	0DE5H	WRA1/WRA2	WRA1

This table is of extreme value to the Assembly Language programmer,since carrying out any arithmetic is as simple as loading the values into the source registers as shown for the particular routine on the previous page and CALLING the routine.The result can then be obtained at the destination as shown on previous page also. One important point when working with WRA1 is the type flag-location 30895 (78AFH).This should always contain a value indicating the type of number currently in WRA1 according to:

02-Integer, 03-String, 04-Single Precision, 08-Double Precision

If an overflow occurs when using the integer routines,the result is converted to single precision and stored in WRA1(the type flag being updated accordingly).

Next issue we look at the math function routines and the 'king' of the routines,the Expression Evaluator.
Meantime,try using some of the routines from this instalment.

DISK DIRECTORY PRINTER

```
5 REM ->-> "DIR"  K.WRIGHT, V.2
10 CLEAR1000:POKE30862,80:POKE30863,52:CLS
15 PRINT"     ======================"
20 PRINT"     < DIRECTORY  PRINTER >"
25 PRINT"     ======================"
30 IF INKEY$<>"" THEN GOTO 30
35 PRINT@260,"DO YOU WANT PRINTOUT ?":PRINT TAB(13);"(Y/N)"
40 K$=INKEY$:K$=INKEY$:IF K$="" THEN GOTO40
45 IF INKEY$<>"" THEN GOTO 45
50 IF K$="N" THEN HC=0:GOTO 60
55 IF K$="Y" THEN HC=1 ELSE GOTO10
60 PRINT@0,"  ":Z=PEEK(28672):CLS:DIR
65 A$=CHR$(PEEK(28672)):POKE 28672,175
70 FOR N=28673 TO 29183
75 IF PEEK(N)=Z AND ASC(RIGHT$(A$,1))=ZTHEN85
80 A$=A$+CHR$(PEEK(N)):POKE N,175
85 NEXT
90 B=0:CLS
95 DIM B$(30):A$=A$+"  "
100 FOR N=1 TO LEN(A$)-2
105 X$=MID$(A$,N,2)
110 IF ASC(RIGHT$(X$,1))=122 THEN B=B+1:SOUND31,1:GOTO125
115 S=USR(S)
120 B$(B)=B$(B)+MID$(A$,N-1,1)
125 NEXT
130 PRINT"          DISC  DIRECTORY":PRINT
135 FOR N=1 TO B
140 PRINT USING"##";N;:PRINT" ";B$(N),
145 NEXT
150 IF HC=0 THEN GOTO 175
155 LPRINT CHR$(14);"    DISC  DIRECTORY":LPRINT
160 FOR N=1 TO B
165 LPRINT USING"##";N;:LPRINT" ";B$(N),
170 NEXT
175 REM >>>>>>>> SELECT ROUTINES START HERE
180 REM    >>>>> USE YOUR OWN PROGRAM TO SELECT DESIRED PROG.
185 PRINT:STATUS
187 PRINT@420,"USE'A'KEY TO RUN AGAIN"
190 I$=INKEY$:K$=INKEY$
195 IFK$="A"THENGOTO10ELSE190
200 END
```

Dick Smith Electronics

Technical Bulletin

Printing Lower Case and Control Characters on the VZ-200/300

As you may be aware the VZ-200/300 computers do not display the standard lower case ASCII characters. Instead they display inverse and graphics characters. To obtain a printout of these special characters the BASIC switches the printer to the graphics mode, outputs a graphic image of the non-standard character. Then switches the printer back to the text mode (NOTE. the X-3250 GP-100 printer is the only printer that the special characters can be printed on).

Because the BASIC used in the VZ-200/300 filters certain ASCII characters when the LPRINT or LLIST statements are used when you wish to print one of these characters in their standard form an OUT statement must be used. Here is a program lising to show the use of the OUT statement.

```
10 FOR X= 1 33 TO 127
20 OUT 1,X
30 OUT 2,X
40 FOR T= 1 TO 100:NEXT T
50 NEXT X
60 OUT 1,0
70 OUT 2,0
```

There are three points to notice in the above program

1. You must OUT to both ports 1 and 2 in that order.

2. Because the OUT statement does not check the status of the port you must include a delay loop (see line 40) to allow the printer time to actually print each character.

3. Because the OUT statement can disturb the BASIC stack pointers you must reset the parallel port before using the LPRINT or LLIST statements. This is done with the lines 60 & 70.

It is important to know that the VZ-200/300 computers use the ASCII character 00 (NUL) to reset the parallel port. This means that printers that use the ASCII 00 as a control character will have some problems in operation on the VZ-200/300 computers. In most cases (but not in all) printers that use the ASCII 00 have an alternate character that can be used (usually ASCII 128).

P/S B Beaver

Paul Beaver

No 114

Dick Smith Electronics

Technical Bulletin

OBTAINING COLOUR ON THE VZ300

1. Background colour - this can be either green or buff (pink) and is changed by entering:

```
     COLOR, 0 (for green - default color)
or   COLOR, 1 (for buff)
```

This leaves the foreground colour of graphics characters unchanged. Note also that only these graphics characters which are to be found on the following keys, (while pressing the shift key: Q, W, E, R, T, Y, U, I, A, S, D, F, G, H, J) may be used to change foreground colours. Any others will always remain black.

2. Foreground colour - this can be changed in two ways.

a) Changing the background colour as well:
enter - COLOR I, J
where I is the foreground colour from 1 to 8 (list of colour codes below) and J is the background colour as above.

b) Changing only the foregound colour:
enter - COLOR I
where I ranges from 1 to 8 as in a).

Note: In both a) and b), no changes to foreground colour will be noticed unless followed by a print statement.

COLOUR CODE TABLE

Code	Colour
1	Green
2	Yellow
3	Blue
4	Red
5	Buff (pink)
6	Cyan
7	Magenta
8	Orange

eg:

```
 10   for I = 1 to 8
 20   Color I
 30   Print '<shift QWE>'
 40   Next I
 50   Color, 1
 60   For I = 1 to 8
 70   Color I
 80   Print '<shift UIA>'
 90   Next I
100   Color 2, 1
110   Print '<shift GHJ>'
```

Regards -
David Summers

SOFTWARE REVIEW

DATA

PROGRAM WRITTEN BY ROBERT QUINN

"DATA" is best discribed as a expandable Diskbased Database. Robert has been able to put together a very useful Database program which is full of features you would expect to find in a more expensive program.

A Database is a program written to handle and store information.It is madeup of pages as you would have in a file or folder.

In "DATA" each page of this file will hold ten records and each record can be upto thirty characters long.

"DATA" is special in that it is expandable,in as far as how many pages and records you can have in your file.
If you have a VZ300 then your Datafile will have 190 records organised on 19 pages.
On the other hand if you had a 16k memory module installed,then you will have 740 records on 74 pages,which is heap's of room for your records.

"DATA" also has all the usual features of a Database you would expect to find,such as the ability to sort or edit your Datafile.
You can also save/load and make a hardcopy of your information.

If you want value for money and a way to store all your information then "DATA" is the program for you.
The cost of A$20.00 makes it one of the cheapest information storage and retrieval mediums you can buy.

My version has no instruction manual as such.But there are 3 datafiles on the Disk which,when loaded,will give you instructions on how to use "DATA".This can be a problem sometimes if you get stuck in the middle of a file and need help.

It's not a big problem and is easy overcome if you make a hardcopy of the instructions.
"DATA" would be one of the most useful and value packed programs available to do what is a important function of most computer system's.
I use this program all the time and find it superb to store my club records on.

CONCLUSION

Well written program. Very useful and good value.

Review written by PETER HILL (Auckland)

AUSTRALIAN VZ USERS GROUPS

VZ USER---Mark Harwood P.O Box 154 Dural N.S.W 2158

LE'VZ OOP---John D'Alton 39 Agnes St. Toowong Qlnd 4066

VZ DOWN UNDER---Scott Le Brun 59 Brentwood Dve Wantirna 3152

HUNTER VALLEY VZ USERS Group---P.O BOX 161 Jesmond N.S.W 2299

WAVZ---Graeme Bywater P.O BOX 388 Morley W.A 6062

ROBO COP.

WRITTEN BY DANNY TODD

```
10 CLS:H=H:GOSUB30
20 SP=0.3:H=H:K=0:S=0:L=0:GOTO140
30 PRINT@73,"▀▀▀ROBOCOP▀▀▀":PRINT
40 PRINT@105,"▬▬▬▬▬▬▬▬▬▬▬▬"
50 PRINT:PRINT"  HELP ROBOCOP FIND HIS WAY"
60 PRINT"    WATCH FOR THE DIRECTIONS"
70 PRINT"    AND REPEAT THEM FOR HIM"
80 PRINT:PRINT"   USE THE <▌> AND/OR <▐> KEYS"
90 PRINT:PRINT
100 PRINT"        PRESS 'P' TO PLAY"
110 B$=INKEY$:C$=INKEY$
120 IFC$<>"P"THENGOTO110
130 RETURN
140 CLS:FORQ=0TO3:PRINT:NEXT
150 IFK=5THENL=L+1:K=0
160 M$=""
170 C$=""
180 FORT=1TOL
190 N=RND(0)
200 IFN>=0.5THENPRINT"                    ▀▀▀▀":GOSUB470:M$=M$+"L"
210 IFN<0.5THENPRINT"                    ▀▀▀▀":GOSUB470:M$=M$+"R"
220 NEXTT
230 K=K+1
240 FORD=1 TO 200*L*SP
250 B$=INKEY$:C$=INKEY$:C$=INKEY$
260 NEXTD
270 CLS:PRINT:PRINT
280 E=0
290 PRINT"WHAT WERE THE DIRECTIONS"
300 FORT=1TOL
310 PRINT"?"
320 B$=INKEY$:C$=INKEY$:C$=INKEY$
330 IFC$<>"R"ANDC$<>"L"THENGOTO320
340 PRINTC$
350 IFC$<>MID$(M$,T,1)THENE=1
360 NEXTT
370 IFE=0THENS=S+1
380 IFS>=HTHENH=S
390 IFE=1THENPRINT"YOU LOST HIM ! YOU'RE FIRED !!"
400 PRINT"SCORE=";S
410 PRINT"HIGH SCORE=";H
420 FORW=0TO1000:NEXT:IFE=1THENGOSUB460:GOTO10
430 FORD=1TO500*SP
440 NEXTD
450 GOTO140
460 FORW=1TO3000:NEXT:CLS:RETURN
470 FORW=5TOINT(RND(15)):NEXT:RETURN
```

P.O.BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USER'S CLUB 1986

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT..
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT THEY ARE FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR THE BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME BETWEEN 6.30 PM AND 9 PM ----PHONE 673-967

HOW TO WRITE ADVENTURE GAMES

WRITTEN BY SCOTT LE BRUN

You all must know that ADVENTURE games are great fun to play,but have you ever tried to write one?.
Most of those who have tried,didn't get far before giving up in disgust!.

An ADVENTURE program is a kind of Interactive database,where you move through the land or building changing the data stored in memory as you do things in your own world that you've created.

Most ADVENTURE games require you to type in English words to act out what you wish to do or have done.Since the computer wouldn't have a clue about understanding English,we have to do a bit of thinking and complex programming.

To Start.

The first thing to do is draw yourself a map of your scenerio,or layout.
This is best done on a grid.We'll start with a 8x8 grid.Number the squares from 0 to your largest number,(use 63 if using a 8x8 grid).
Number all squares from left to right and insure all squares are numbered.
Next write In all the names of your locations and the routes or exits out of them.
Draw in lines along the squares where walls are to be..IE

Square 1
'Over Grown Archway'
Exits:-'N,S,E'To the west is a wall.

Now think up a list of all the treasures that your hero is to find.Also any objects that can be used to help him/her in his/her quest.
It is a good idea to make them relevent to your scenerio.Swords and magic amulets are good for fighting ancient evil back in the dark ages.While a kill-o-zap gun and matter transporters might be useful when fighting off a galactic menace from taking over planet earth near the end of the 21st century.
You also need to note in which rooms or locations that certain objects are to be located or hidden.

Now work out all your traps (and solutions!) and list them down.
Some traps could be:-
Plugging a control panel into a matter transporter for it to work, possibly to get out of a dangerous situation.

Saying a magic word to open a hidden doorway.

Another important thing you have to remember is what words you want the computer to recognise in regard to moving around and manipulating objects.This should be done using the 'VERB-NOUN format.
IE..SWING SWORD,FIRE GUN,GO NORTH,TURN KEY or even CLIMB UP TREE.

Now that you have the whole lot down on paper,Next issue we'll look at how to convert all your list's and the map into your very own ADVENTURE program.

Continued next issue.

As I was getting quite a few disk I/O errors I decided to give the head a clean using a CLEANMASTER head cleaning disk purchased from Dick Smith. The head is nice and clean now, but my drive no longer READS or WRITES anymore and I have'nt had chance to fix the problem as I had a deadline to meet with this issue. I suspect the pressure pad may have been damaged. My second drive is OK and I'm thankful it did'nt suffer the same fate as it's head was cleaned first.

The reason for the mishap is quite simple, it's lack of proper instructions that come with the disk. The cleaning disk is designed for dual head drives and as the VZ drive has only a single head it's quite easy to put it in the wrong way and that's what I did. The instructions say to insert the C/Disk according to the arrows, but there's no arrows on disk. For VZ drives use the following steps :-

1) Turn C/Disk over and put no more than 5 (FIVE) drops of cleaning fluid to both oblong cutouts on back of disk.

2) Turn disk right way up again so that the label with instructions is on top.

3) Turn disk 90 Deg. so that the label is on the left side of disk and the WRITE PROTECT notch is away from you.

4) Insert C/Disk in drive and close drive door.

5) Type in DIR and press RETURN. This will start the cleaning process.

6) After 30 Sec's press (-) key to stop drive. Open drive door and remove C/Disk.

Below are a series of drawings showing the CLEANMASTER disk for persons not familiar how the C/Disk is used by the drive.

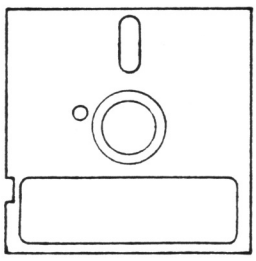
The drawing on the left depicts top side of C/Disk while the one next to it shows reverse side of C/Disk. Note the three cutouts on this side.

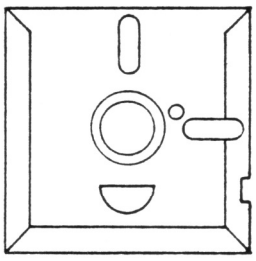
The third drawing shows the drive recording head, inserted disk and pressure pad. With the drive door open the pressure pad is lifted up allowing for easy removing/inserting of disks.

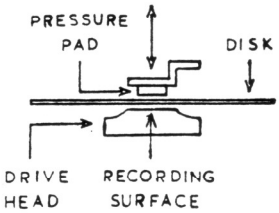
The C/Disk must be inserted into drive as shown in the last drawing on the right. The reason for that is because the C/Disk is made of different material and has a rougher surface. If inserted the normal way the pressure pad would rest on the cleaning surface and there would be a good possibility of the pressure pad tearing the cleaning surface which is what happened to me.

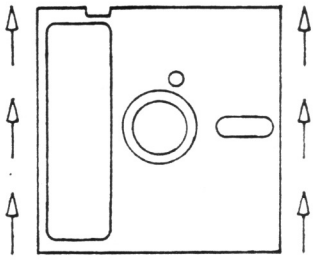
The act of closing drive door lowers pressure pad onto the exposed surface of the disk in the cutout pressing disk against the head where reading/writing from/to disk is done. Note :- READING/WRITING is done on bottom of disk.

The comments above relate to the CLEANMASTER HEAD C/DISK only. If using other brands of C/Disk make sure that the cleaning surface is on bottom and that the pressure pad does not rest on any part of cleaning surface.









SOFTWARE FOR SALE

Prices shown include postage and packing anywhere in Australia. New Zealand customers, please add A$2.00 to the prices below. Make sure that the memory of your VZ system is greater than or equal to the memory required by the software that you wish to purchase. Memory sizes indicated do not include Video Ram. Prices shown are for both tape and disk copies All prices are in Australian dollars.

* PLEASE NOTE *

SNAKE DUEL & STELLAR WARS have both been dropped from the range due to lack of sales.

VZ MONOPOLY V2.0 Both the 22K and 32K versions are included together, so that if you have a 22K system and have the 22K VZ MONOPOLY V2.0 and you wish to upgrade to the 32K version, you don't have to re-purchase the same program.

NEW SOFTWARE

FBI 2001

Welcome to the year 2001.
You as a future agent of the FBI have been called forth to capture a criminal within a certain time limit. Using your teleport, you track the crook around the world picking up clues to his identity and his next destination as you go. With this information you must get a warrent from HQ before you can legally arrest him or her.

After finishing a case you go up in the ranks. The ultimate rank is FBI-ELITE!

Overall an excelent program, that uses lo-res graphics to draw scenes of each location. Written by P.Frantz.

BLACKJACK ROYALE	- 16K	- $12.50
CASTLE GREYSTONE	- 32K	- $12.50
DATA	- 16K	- $20.00
FBI-2001	- 32K	- $15.00
GALACTIC EMPIRES	- 16K	- $12.50
GRAPHSTAR	- 22K	- $20.00
HAUNTED MANSION	- 22K	- $12.50
KNIGHTS QUEST	- 32K	- $15.00
NAMESTAR/SHOPSTAR	- 22K	- $12.50
SCOTLAND YARD	- 16K	- $6.50
TRIVIAL CULT	- 16K	- $6.50
VZ MONOPOLY V2.0	- 22K	- $15.00

ADVENTURER'S COLLECTION - Contains SCOTLAND YARD, HAUNTED MANSION, KNIGHTS QUEST & CASTLE GREYSTONE.

All on the one tape or 2 disks - $36.50

Please make all cheques and money orders payable to :-

Scott Le Brun
59 Brentwood Dve
Wantirna VIC 315